

**NAME**

`ne_malloc`, `ne_calloc`, `ne_realloc`, `ne_strdup`, `ne_strndup`, `ne_oom_callback` - memory allocation wrappers

**SYNOPSIS**

```
#include <ne_alloc.h>
```

```
void *ne_malloc(size_t size);
```

```
void *ne_calloc(size_t size);
```

```
void *ne_realloc(void *size, size_t len);
```

```
char *ne_strdup(const char *s);
```

```
char *ne_strndup(const char *s, size_t size);
```

```
void ne_oom_callback(void (*callback)(void));
```

**DESCRIPTION**

The functions `ne_malloc`, `ne_calloc`, `ne_realloc`, `ne_strdup` and `ne_strndup` provide wrappers for the equivalent functions in the standard C library. The wrappers provide the extra guarantee that if the C library equivalent returns NULL when no memory is available, an optional callback will be called, and the library will then call `abort()`.

`ne_oom_callback` registers a callback which will be invoked if an out of memory error is detected.

**NOTES**

If the operating system uses optimistic memory allocation, the C library memory allocation routines will not return NULL, so it is not possible to gracefully handle memory allocation failures.

**AUTHOR**

**Joe Orton** <neon@lists.manyfish.co.uk>

Author.

**COPYRIGHT**